ALTOII DOCUMENTATION

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

INTRODUCTION

THE ALTO

The Alto is used to prepare and print documents containing text, diagrams, and images, to convey messages electronically, to aid in circuit and IC design, and to write programs. It is a minicomputer consisting of a processor, disk drive, workstation, and Ethernet transceiver. Optionally, it may be ordered with a Diablo Hytype printer and/or a second disk drive. The Alto can also be used to drive other devices ranging from communications interfaces to production machines.

There isn't much special-purpose hardware in the Alto. Most of what you read about in the hardware manual is in fact implemented by microcode. This gives us considerable flexibility in the way we design software interfaces for experimental devices and specialized instructions.

A standard Alto system includes:

An 875-line television monitor, oriented with the long tube dimension vertical. This monitor provides a 606 by 808 point display which is refreshed from main memory at 60 fields (30 frames) per second. One bit in memory shows up as one bit on the screen. Since the screen is 606 by 808 points, a quick calculation shows that a full-screen display requires nearly 32K of the Alto memory. For a machine with only 64K of memory minimum, that seens a big price to pay. The theory is that in exchange for the space we get enormous freedom to experiment with various strange ways of manipulating the screen. It has programmable polarity, and a cursor, whose position is controlled by the mouse, and is a 16 by 16 bit-map whose shape is under program control, independent of display content.

An encoded keyboard.

A mouse (pointing device).

A Diablo Model 31 disk drive. The Model 31 accepts a single disk which can be used to store about 2.5 megabytes. The average seek time is 70 ms, the average transfer rate, 1.22 MHZ. A second drive can be added.

An interface to the Ethernet, a 3mbps serial communication line. An Ethernet transceiver connects the Alto to the Ethernet. Using the Ethernet, an Alto can communicate with a large number of other Altos, computers, special purpose servers.

The microcoded processor has 64K of 850ns, 16-bit word semiconductor memory (expandable up to 256K in 64K banks). A 1K or 3K microinstruction Ram which can be loaded with special purpose microcode to extend the instruction set, perform special functions or drive special I/O devices.

The processor, disk, and their power supplies are packaged in an under-table-size cabinet for easy placement in the user's office. The other I/O devices may be a few feet away, and are packaged for desk top use.

THE NETWORK

The network is composed of Altos and other computers connected to several geographically dispersed, technologically innovative local computing nets which, in turn, are tied together by minicomputers over standard leased and/or dial-up telephone lines. The local computing nets are called Ethernets: the minicomputers linking them are referred to as Gateways, the latter providing several services in addition to linking together Ethernets.

Question: Why have a network? Answer: Because it's nice to be able to pawn off some of the work on other machines, leaving your Alto free. That's why we have a number of machines generically called servers. Normally, server Altos have special purpose, expensive hardware attached to them (e.g. large-capacity disks, printers), and their sole purpose in life is to make that hardware available to more than one person/Alto. We tend to identify servers by function, so we talk about printing servers, file servers, name lookup servers, mailbox servers, and so on. Many of the protocols for use on the Ethernet were developed precisely so that personal Altos could communicate effectively with server Altos.

We all know how uncommunicative computers can be, and Altos are no different. That's why we invent careful protocols for them to use in talking to each other. Most of the protocols now in use on the Ethernet are called Pup-based (Parc Universal Packets). You will probably hear some of the following protocol names being tossed about in conversation:

(0) EEFTP- a grand-daddy of file transfer protocols (Experimental Ether File Transfer Protocol). No longer in active use.

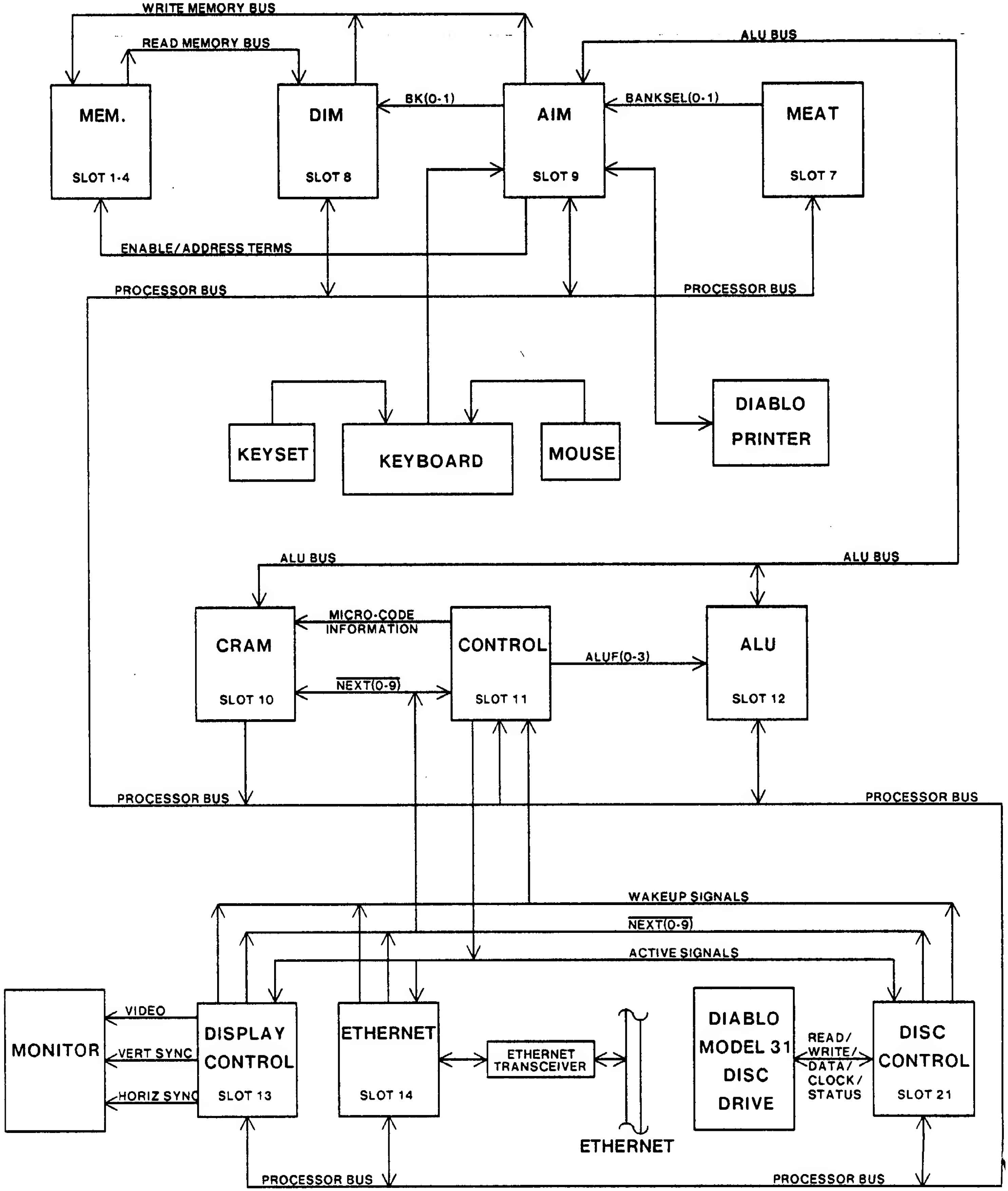
(1) EFTP- stands variously for Early FTP, Ears FTP, Experimental FTP. A venerable protocol now mostly used to transfer files to printing servers. The Alto program Empress uses it for this purpose.

(2) FTP- refers to File Transfer Protocol, as well as the Alto that implements it and provides an interactive user interface. If you come from the Arpanet world. Don't confuse this FTP with the one out there - ours is Pup-based and incompatible. On MAXC, where both the Pup and Arpa FTP protocols come in handy, the name FTP refers to the Arpa one and PupFTP (obviously) refers to the Ethernet one.

(3) BSP- the Byte Stream Protocol. Built on top of Pup, this protocol is used by conversants who want to view the network as a full-duplex stream of 8-bit bytes. BSP is used to implement FTP.

(4) MTP- the Mail Transfer Protocol. Used by Laurel (the Alto-based message system) to ship messages to and from mailbox servers.

ALTO II MODULE BLOCK DIAGRAM



BOOTING INFORMATION

The process of "booting" the Alto is one of setting some or all of the Alto's state either by reading a file from the disk or by accepting packets from the Ethernet.

There are four basic steps in "booting" the Alto: (1) the tasks in the microprocessor are reset; (2) a 256-word "boot loader" is loaded into main memory and started; (3) the boot loader loads a portion of Alto main memory from a "boot file" and finishes by transferring to a known place; (4) the user's program loaded by the third step can restore even more of the Alto's state.

Booting

"Booting" is accomplished either by pushing the "boot button" located on the rear of the keyboard or by the Reset Mode Register. the emulator task is started in a standard boot program. This program reads location #177034, a word whose contents can be altered by pushing various keys on the keyboard. If the <bs> & <quote> key is depressed during booting, the machine state will be restored from the Ethernet: otherwise, the state is restored from the disk.

When booting from the disk, the keyboard word is interpreted as a disk address where a "disk boot loader" is located. If no keys are depressed, disk address 0 is generated, which is the normal resting place of the "disk boot loader" for the operating system. The emulator reads a single 256-word disk record into memory locations 1, 2, ...#400: the 8-word disk label for this page is placed in #402, #403, ...#411. When the disk transfer is complete, control is transferred to location 1 in the loader. The boot loader uses the saved label to point to the remainder of a "boot file" which is read into main memory and started.

When booting from the Ethernet, the microcode waits until a "breath of life" packet arrives, containing a 256-word "ethernet boot loader" which is read into locations 1-#400 and executed by transferring to location 3. It is up to this loader to establish communications with a party willing to deliver the remainder of the state needed.

ALTO II MODULE LOCATION CHART

SLOT	MODULE NAME	VENDOR P/N
1	MEMORY STORAGE	216273
2	MEMORY STORAGE	216273
3	MEMORY STORAGE	216273
4	MEMORY STORAGE	216273
5		
6		
7	TERMINATOR	216421
8	MEMORY DATA INTERFACE	216312
9	MEMORY ADDRESS INTERFACE	216347
10	CONTROL RAM (CRAM)	216365
11	2K CONTROL MODULE	216484
12	ARITHMETIC LOGIC UNIT (ALU)	216381
13	DISPLAY CONTROL	216339
14	ETHERNET	216323
15	TRIDENT	
16		
17	ORBIT CTL	
18	INP.	
19	MEM	
20	OUT	
21	DISC CONTROL	216389

ALTO II EXTENDED MEMORY MODULE LOCATION CHART

SLOT	MODULE NAME	VENDOR P/N
1	MEMORY STORAGE	217188
2	MEMORY STORAGE	217188
3	MEMORY STORAGE	217188
4	MEMORY STORAGE	217188
5		
6		
7	MEAT	216646
8	MEMORY DATA INTERFACE	217174
9	MEMORY ADDRESS INTERFACE	217173
10	CONTROL RAM (CRAM)	217176
11	2K CONTROL MODULE	217175
12	ARITHMETIC LOGIC UNIT (ALU)	216381
13	DISPLAY CONTROL	216339
14	ETHERNET	216323
15		
16		
17		
18		
19		
20		
21	DISC CONTROL	216389

ALTO II EXTENDED MEMORY 3KCRAM MODULE LOCATION CHART

SLOT	MODULE NAME	VENDOR P/N
1	MEMORY STORAGE	217188
2	MEMORY STORAGE	217188
3	MEMORY STORAGE	217188
4	MEMORY STORAGE	217188
5		
6		
7	MEAT	216646
8	MEMORY DATA INTERFACE	217174
9	MEMORY ADDRESS INTERFACE	217173
10	3K CONTROL RAM (3KCRAM)	217812
11	3K CONTROL MODULE	218094
12	ARITHMETIC LOGIC UNIT (ALU)	218090
13	DISPLAY CONTROL	216339
14	ETHERNET	216323
15		
16		
17		
18		
19		
20		
21	DISC CONTROL	218092

NOTE 1: EXCEPT MEMORY STORAGE AND 2K CONTROL MODULES IN EXTENDED MEMORY MACHINES, THE ALU AND DISC CONTROL MODULES IN ALL MACHINES, UNLESS MODULES HAVE THE SAME VENDOR P/N's THEY ARE NOT INTERCHANGEABLE.

NOTE 2: EXTENDED MEMORY ALTO II AND EXTENDED MEMORY 3KCRAM ALTO II MACHINES HAVE DIFFERENT VENDOR P/N's FOR THE MEMORY STORAGE MODULES. THESE P/N's ARE NOTED BELOW FOR THE DIFFERENT SIZES OF MEMORY.

MEMORY SIZE	VENDOR P/N
64 K	217187
128 K	217188
256 K	217189

NOTE 3: THE 2K CONTROL MODULE IN THE EXTENDED MEMORY ALTO II MACHINES MAY HAVE THE FOLLOWING P/N's, 217175 OR 218147.

+15, -15, +12 VOLT POWER SUPPLY INFORMATION

GENERAL

SSD units are equipped with two controls; the output voltage, and the overvoltage adjustment (OVP) potentiometers accessible at the terminal-board (TB1) end of the supply. The output VOLT control (R26) varies the output voltage while the OVP control (R24) sets the OVP trip point. Both are facroty-set to nominal values.

OVERVOLTAGE (OVP) TRIP POINT

The OVP trip point is factory-set to 1.2 volts or 10% (whichever is greater) higher than the rated nominal output voltage.

OVP TRIP POINT ADJUSTMENT

To adjust the OVP, proceed as follows:

1. Rotate OVP panel control R24 fully CW.
2. Rotate output VOLT control R26 on panel until the output voltage is equal to the desired trip point.
3. Rotate OVP control R24 slowly CCW until the output voltage suddenly drops to zero. This indicated that the OVP circuit has been triggered.
4. Remove the input power. Rotate VOLT control several turns CCW. Allow approximately 10 seconds for unit to discharge.
5. Apply input power. Reset output to desired operating voltage.

NOTE:
The OVP circuit includes a time-delay network such that the overvoltage condition must exist for approximately 100 microseconds before the OVP fires. This delay prevents short-duration overvoltage pulses from triggering the circuit.

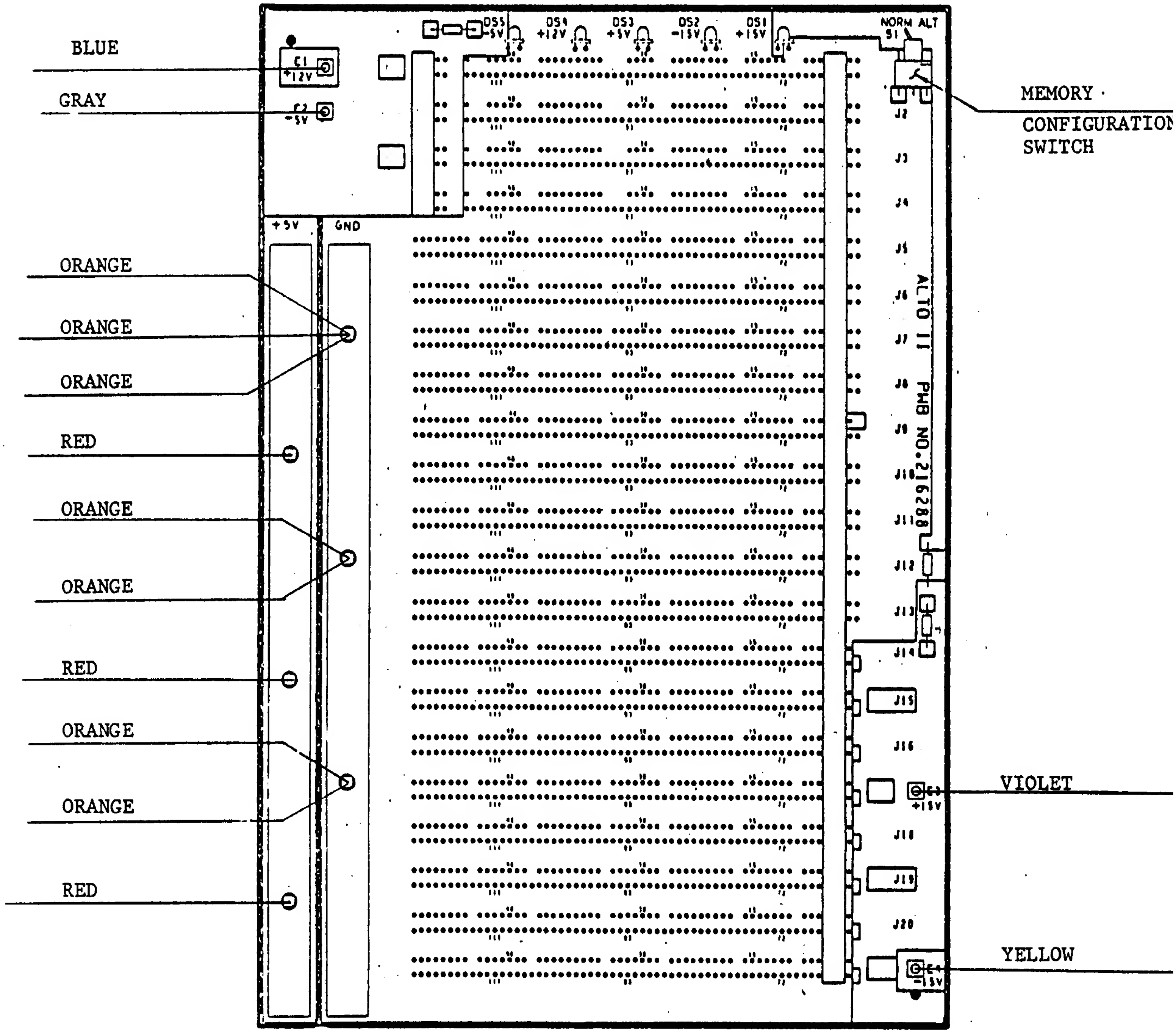
RESET AFTER OVP FIRES

If the OVP fires, proceed to reset the circuit as follows:

1. Remove input power and disconnect load (in case overvoltage condition is externally induced). Allow approximately 10 seconds for unit to discharge.
2. Rotate VOLT adjustment R26 fully CCW (minimum voltage).
3. Apply input power and raise output voltage to desired value.

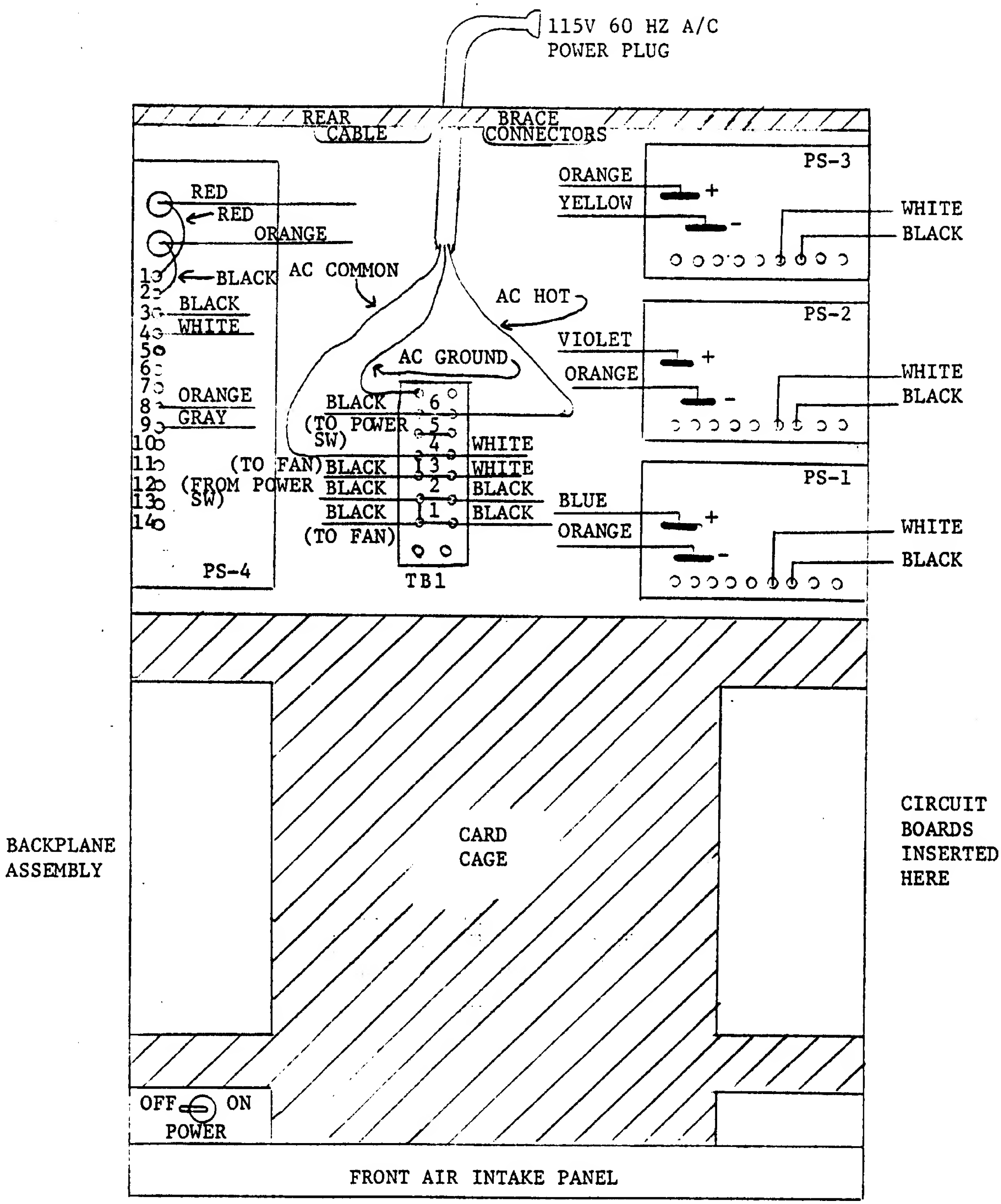
NOTE:
If the OVP trips again, internal failure is indicated, or the output is set too close to operating voltage.

TOP



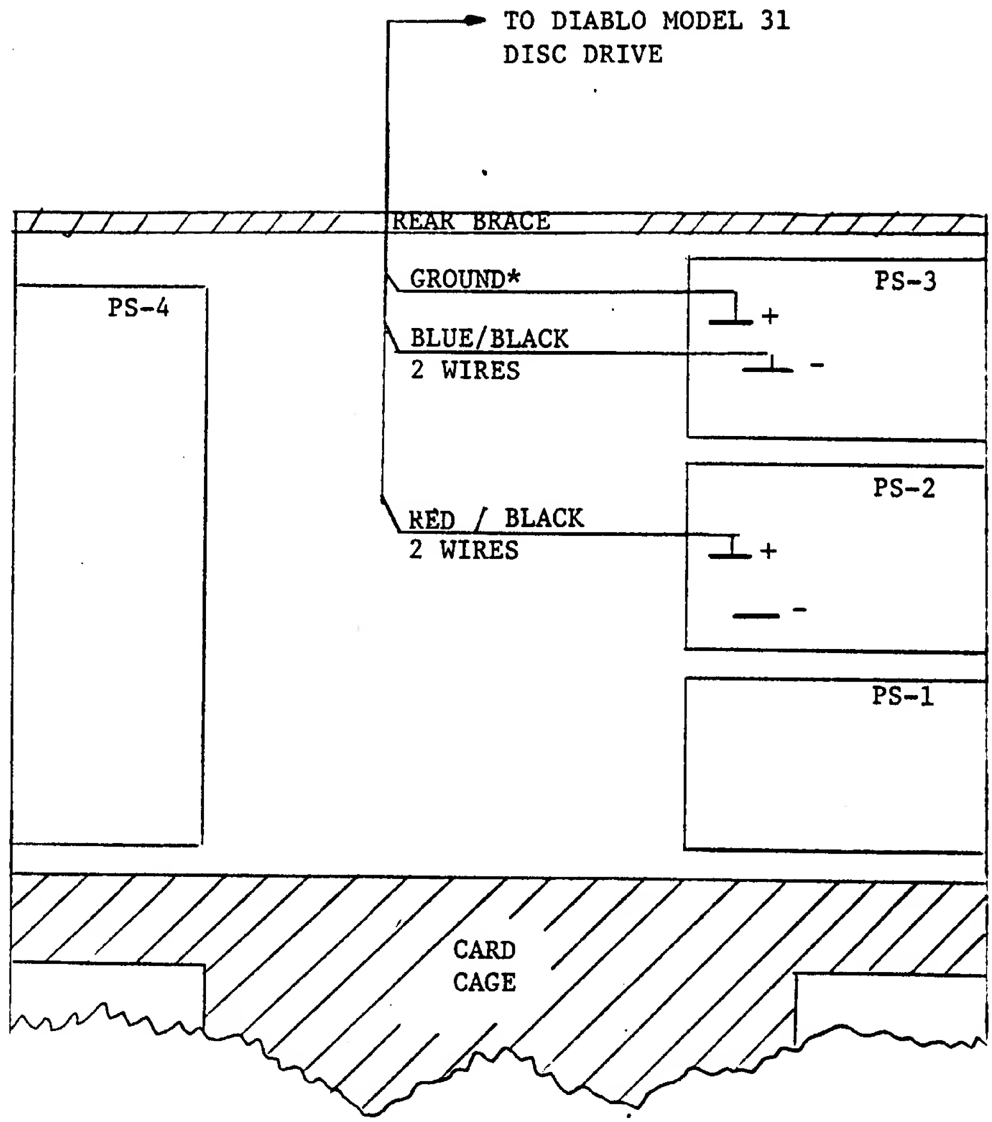
WIRE-WRAP SIDE

BACKPLANE ASSEMBLY



PS-1 = +12 VOLT POWER SUPPLY
PS-2 = +15 VOLT POWER SUPPLY
PS-3 = -15 VOLT POWER SUPPLY
PS-4 = $\pm$5 VOLT POWER SUPPLY

ALTO II POWER WIRING DIAGRAM
(WITHOUT DISC DRIVE CONNECTIONS)



*NOTE: GROUND WIRE MAY BE EITHER A BARE WIRE OR A BARE WIRE WITH BLACK SHRINK TUBING ON IT.

PS-1 = +12 VOLT POWER SUPPLY
PS-2 = +15 VOLT POWER SUPPLY
PS-3 = -15 VOLT POWER SUPPLY
PS-4 = ±5 VOLT POWER SUPPLY

ALTO II POWER WIRING DIAGRAM

FOR DIABLO MODEL 31

DISC DRIVE

ALTO II PREVENTIVE MAINTENANCE OUTLINE

1) Remove power from Alto II system.

2) Clean and/or replace filter in front of Processor cabinet.

3) Check for smooth slide action of Processor chassis.

4) Verify ribbon cables are properly seated on modules.

5) Verify ribbon cables are firmly attached to Processor inside frame.

6) Check ribbon cables for cracks, pinched, or other abnormalities.

7) Verify bulkhead cable connectors are properly seated.

8) Switch the position of the memory configuration switch.

9) Check A/C input for correct polarity.

10) Reapply power to Alto II system.

11) Check fans (4) for proper operation.

12) Check and adjust (if necessary) power supplies: +15v, -15v, +12v, +5v, -5v.

13) Clean and/or replace air filter in front of disc drive.

14) Check for smooth slide action of disc drive.

15) Verify disc drive cables are not being pinched.

16) Verify smooth loading and unloading of disc packs.

17) Clean disc drive R/W heads.

18) Check and adjust disc drive Track Zero adjustment.

19) Check and adjust disc drive R/W head adjustment.

20) Check and adjust disc drive Sector Transducer adjustment.

21) Run Diex.run (using scratch pack) to verify proper operation of disc drive.

22) Verify Data Gate and Write Current are correct.

23) Verify proper operation of Processor by running Madtest.run. (Maybe run while doing disc drive adjustments.)

24) Verify proper operation of Keyboard by using Keytest.run.

25) Clean inside of mouse for smooth operation .

26) Verify proper operation of Mouse by using Keytest.run

27) Clean Display screen.

28) Adjust Display screen as required for optimim presentation using Crttest.run.

29) Verify proper operation of Processor by running Ramload.run.

30) Verify proper operation of Ethernet by running Puptest.run and/or Edp.run.

31) Verify proper operation of the overall system by running BFSTEST.RUN.

32) Verify proper operation of the overall system by running the following software programs: Neptune.run, Bravo.run, Scavenger.run.

33) Verify proper operation of Memory by running Dmt.boot.
Note: The machine should be left running Dmt.boot.